

CLAIMS

I claim:

1. A system for collecting performance data by an operating system that monitors a performance of a target application executing on a computer system, the system comprising:

a data structure maintained by the operating system for storing at least performance data of the target application collected by the operating system;

a virtual memory mapping that maps the data structure to a monitoring application allowing the monitoring application to read the performance data from the data structure without transferring the performance data to the monitoring application using a system interrupt.

2. The system as set forth in claim 1 further including one or more communication parameters maintained in the data structure which are set to pass information between the operating system and the monitoring application without a system call.

3. The system as set forth in claim 2 wherein the one or more communication parameters include one or more parameters predefined to control handshaking between the operating system and the monitoring application.

4. The system as set forth in claim 2 wherein the one or more communication parameters include a full flag that when set, instructs the monitoring application to read the performance data from the data structure.

5. The system as set forth in claim 2 wherein the one or more communication parameters include at least one of: one or more bits, one or more counters, and one or more data registers.

6. The system as set forth in claim 1 wherein the data structure includes one or more memory buffers, one or more memory locations, one or more data registers, or a combination of each.

7. The system as set forth in claim 1 further including a plurality of data structures maintained by the operating system to store at least the performance data.

8. The system as set forth in claim 1 further including a performance monitoring unit in communication with the operating system that collects the performance data of the target application and loads the performance data into the data structure.

9. A system of communicating between a kernel and an application in a user space, the system comprising:

a data structure maintained by the kernel for storing data, the data structure being virtually addressed to the application;

one or more parameters associated to the data structure that represent one or more predetermined communications where the kernel and the application communicate by changing the one or more parameters to reduce communication by system calls.

10. The system as set forth in claim 9 further including header information defined within the data structure, the header information containing the one or more parameters.

11. The system as set forth in claim 9 wherein the one or more parameters include a read parameter that is changeable by the kernel to indicate to the application that the application may read the data stored in the data structure.

12. The system as set forth in claim 9 wherein the data structure is a data buffer.

13. The system as set forth in claim 9 wherein the one or more parameters include one or more values to control handshaking between the kernel and the application.

14. The system as set forth in claim 9 further including at least one of memory addresses, counters, and data registers that store the one or more parameters.

15. The system as set forth in claim 9 wherein the one or more parameters are virtually mapped to the application allowing direct access thereto.

16. A method of transferring data between an operating system and a user application, the method comprising the steps of:

defining a data structure in the operating system for storing data, the data structure being mapped to one or more addresses controlled by the operating system;

virtually mapping the data structure to the user application providing at least read access to the data structure;

storing data, by the operating system, in the data structure; and

reading the data, by the user application, from the data structure through the virtual mapping.

17. The method as set forth in claim 16 further including collecting performance data of a target application and loading the performance data into the data structure.

18. The method as set forth in claim 16 further including passing instructions between the operating system and the user application using the data structure without a system call.

19. The method as set forth in claim 18 wherein the passing instructions includes setting one or more parameters within the data structure that represent predefined instructions.

20. The method as set forth in claim 18 wherein the passing instructions includes including setting a read flag in the data structure that instructs the user application to read the data from the data structure.

21. The method as set forth in claim 16 further including controlling handshaking between the operating system and the user application by setting parameters within the data structure that are readable by the user application.

22. The method as set forth in claim 16 further including defining an overflow data structure for storing data when the data structure is full.

23. The method as set forth in claim 16 further including mapping the data structure to the one or more addresses controlled by the operating system by virtual memory addressing, physical memory addressing or a combination of both.

24. A computer readable product stored on a computer readable medium comprising:

one or more computer readable instructions that cause a computer to define a data structure within a kernel address space for storing data;

one or more computer readable instructions that cause a computer to virtually map the data structure to an application in a user address space allowing the application to directly read the data from the data structure in the kernel address space.

25. The computer readable product as set forth in claim 24 further including one or more computer readable instructions that cause a computer to define one or more control parameters associated to the data structure where communication handshaking is established between the kernel address space and the user space by setting predetermined values for the one or more parameters.